|  |  |  |  |
| --- | --- | --- | --- |
| **Method to Test: SaveToFile()** | | | |
| **Operation:** | **Purpose:** | **Object State:** | **Expected Result:** |
| **FileOutputStream file = new FileOutputStream(getFileName()); ObjectOutputStream write = new ObjectOutputStream(file);   write.writeObject(tasks);** | **write my LinkedList of tasks to the file** | **fileName = "tasks.txt" task** | **The file that the user selected will be written to containing the linkedList** |
| **FileOutputStream file = new FileOutputStream(getFileName()); ObjectOutputStream write = new ObjectOutputStream(file);   write.writeObject(tasks);** | **generate a FileNotFound exception for the file name the user entered** | **fileName = "someRandomFileThatDoesn'tExist"** | **The method will generate a FileNotFoundException and will instead save the file to a file called "defaultFile.ser". The user will be told the file wasn't found and the default file was written to instead.** |
| **FileOutputStream file = new FileOutputStream(getFileName()); ObjectOutputStream write = new ObjectOutputStream(file);   write.writeObject(tasks);** | **Generate a IOException from having a class not be serializable or a file system error.** | **fileName = "tasks.txt"** | **the method will generate an IOException and will tell the user an error occurred and the file could not be written to.** |
|  |  |  |  |
|  |  |  |  |
| **Method to Test: readFromFile()** | | | |
| **Operation:** | **Purpose:** | **Object State:** | **Expected Result:** |
| **FileInputStream file = new FileInputStream(getFileName()); ObjectInputStream read = new ObjectInputStream(file); tasks = (LinkedList<Task>) read.readObject();** | **Read in my LinkedList from a file named tasks.txt** | **fileName = "tasks.txt"** | **The file's contents will be saved into the tasks LinkedList** |
| **FileInputStream file = new FileInputStream(getFileName()); ObjectInputStream read = new ObjectInputStream(file); tasks = (LinkedList<Task>) read.readObject();** | **generate a FileNotFound exception for the file name the user entered** | **fileName = "someRandomFileThatDoesn'tExist"** | **the method will generate a fileNotFoundException and will instead try and read from a file called "defaultFile.ser". The user will be told the default file was read from instead.** |
| **FileInputStream file = new FileInputStream(getFileName()); ObjectInputStream read = new ObjectInputStream(file); tasks = (LinkedList<Task>) read.readObject();** | **Generate a IOException** |  | **the method will generate and IOException and will tell the user an error occurred and the tasks could not be read.** |
| **FileInputStream file = new FileInputStream(getFileName()); ObjectInputStream read = new ObjectInputStream(file); tasks = (LinkedList<Task>) read.readObject();** | **Generate a ClassNotFoundException from having a task of a class that doesn't exist** | **tasks.get(x).getClass() = UnkownClass** | **The method will generate a ClassNotFoundException and will tell the user that a task was read in of an invalid type.** |
| **FileInputStream file = new FileInputStream(getFileName()); ObjectInputStream read = new ObjectInputStream(file); tasks = (LinkedList<Task>) read.readObject();** | **Generate an EOFException by having the end of the file unexpectedly reached.** |  | **The method will generate an EOFException and will tell the user that the end of the file was reached unexpectedly and the tasks could not be read.** |
| **FileInputStream file = new FileInputStream(getFileName()); ObjectInputStream read = new ObjectInputStream(file); tasks = (LinkedList<Task>) read.readObject();** |  |  |  |